

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently Amended) A method, comprising:  
  
generating a trace for a subset of program instructions formed from a set of program instructions, with said trace comprising a path identifier value, start address, and end address;  
  
storing said trace in a trace buffer;  
  
retrieving said trace from said trace buffer; and  
  
reproducing said subset of program instructions using said trace;  
  
wherein said subset of program instructions comprises a function and a number of traces stored in the trace buffer is proportional to a number of functions in the program instructions.
2. (Original) The method of claim 1, wherein generating said trace comprises:  
  
receiving an endpoint program instruction for said subset of program instructions;  
  
generating said path identifier value and end address for said subset of program instructions;  
  
retrieving said start address from a program counter register; and  
  
generating said trace using said path identifier value, start address and end address.

3. (Original) The method of claim 2, wherein generating said path identifier value and end address comprises:

initializing a path identifier register, with said path identifier register to store said end address and said path identifier value;

assigning each unconditional branch instruction for said set of program instructions an unconditional partial path value and an unconditional offset value; and

assigning each conditional branch instruction for said set of program instructions a taken branch partial path value, an untaken branch partial path value, and a conditional offset value.

4. (Original) The method of claim 3, wherein said generating said path identifier value and end address further comprises:

receiving a branch instruction;

determining whether said branch instruction is a conditional branch instruction or unconditional branch instruction;

incrementing said path identifier value with said unconditional partial path value and said end address with said unconditional offset value if said branch instruction is an unconditional branch instruction;

incrementing said path identifier value with said taken branch partial path value and said end address with said conditional offset value if said branch instruction is a conditional branch instruction that was taken; and incrementing said path identifier value

with said untaken branch partial path value and said end address with said conditional offset value if said branch instruction is a conditional branch instruction that was untaken.

5. (Original) The method of claim 4, further comprising initializing said path identifier register prior to processing another conditional branch instruction.
6. (Canceled).
7. (Original) The method of claim 3, wherein said path identifier and end address each comprise 32 bits.
8. (Original) The method of claim 3, wherein said conditional branch instruction and said unconditional branch instruction each comprise 32 bits.
9. (Previously Presented) The method of claim 3, wherein said conditional branch instruction and said unconditional branch instruction further comprise an operation code value to identify a type of instruction.
10. (Original) The method of claim 1, wherein said trace buffer comprises  $N$  entries, and further comprising moving traces from said trace buffer to another storage location after storing  $N$  entries.
11. (Currently Amended) An apparatus, comprising:

a trace buffer to store traces; and

a trace management module to couple to said trace buffer, said trace management module to further comprise a trace generator to generate a trace for a subset of program instructions formed from a set of program instructions, with said trace comprising a path identifier value, start address, and end address, said trace management module to store each generated trace in said trace buffer, wherein said subset of program instructions comprises a function and a number of traces stored in the trace buffer is proportional to a number of functions in the program instructions.

12. (Original) The apparatus of claim 11, wherein said trace management module further comprises a trace interrupt module to remove traces from said trace buffer.

13. (Original) The apparatus of claim 11, wherein said trace management module further comprises a trace decoder to reproduce said subset of program instructions using said trace.

14. (Original) The apparatus of claim 11, wherein said trace generator further comprises:

a path identifier register;

a program counter register; and

a path identifier generator operatively coupled to said path identifier register and said program counter register, said path identifier generator to receive an endpoint program instruction for said subset of program instructions, to generate said path

identifier value and end address for said subset of program instructions using said path identifier register, to retrieve said start address from a program counter register, and generate said trace using said path identifier value, start address and end address.

15. (Currently Amended) A system, comprising:

a wireless device to process information, said wireless device comprising:

a trace buffer to store traces;

a processor to couple to said trace buffer, said processor to process a subset of program instructions formed from a set of program instructions;

a trace management module to couple to said processor and said trace processor, said trace management module to further comprise a trace generator to generate a trace for said subset of program instructions, with said trace comprising a path identifier value, start address, and end address, said trace management module to store each generated trace in said trace buffer, wherein the subset of program instructions comprises a function and a number of traces stored in the trace buffer is proportional to a number of functions in the program instructions; and

an antenna to communicate said information over radio frequency spectrum.

16. (Original) The apparatus of claim 15, wherein said trace management module further comprises a trace interrupt module to remove traces from said trace buffer.

17. (Original) The apparatus of claim 15, wherein said trace management module further comprises a trace decoder to reproduce said subset of program instructions using said trace.

18. (Original) The apparatus of claim 15, wherein said trace generator further comprises:

- a path identifier register;
- a program counter register; and
- a path identifier generator operatively coupled to said path identifier register and said program counter register, said path identifier generator to receive an endpoint program instruction for said subset of program instructions, to generate said path identifier value and end address for said subset of program instructions using said path identifier register, to retrieve said start address from a program counter register, and generate said trace using said path identifier value, start address and end address.

19. (Currently Amended) An article comprising:

- a storage medium;
- said storage medium including stored instructions that, when executed by a processor, result in generating a trace for a subset of program instructions formed from a set of program instructions, with said trace comprising a path identifier value, start address, and end address, storing said trace in a trace buffer, retrieving said trace from said trace buffer, and reproducing said subset of program instructions using said trace, wherein the subset of program instructions comprises a function and a number of traces

stored in the trace buffer is proportional to a number of functions in the program instructions.

20. (Original) The article of claim 19, wherein the stored instructions, when executed by a processor, further result in generating said trace by receiving an endpoint program instruction for said subset of program instructions, generating said path identifier value and end address for said subset of program instructions, retrieving said start address from a program counter register, and generating said trace using said path identifier value, start address and end address.